UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/757,516 | 01/15/2004 | Craig Hansen | 43876-155 | 4560 |

7590 08/06/2008

McDermott, Will & Emery
600 13th Street, N.W.
Washington, DC 20005-3096

| EXAMINER | |
|---|---|
| COLEMAN, ERIC | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2183 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 08/06/2008 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/757,516 | HANSEN ET AL. | |
| | Examiner | Art Unit | |
| | Eric Coleman | 2183 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *6/27/08,4/18/08*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

## DETAILED ACTION

### *Claim Rejections - 35 USC § 112*

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

The claims 1-18 and 19-32 are rejected under 35 U.S.C. 112, first paragraph as failing to comply with the description requirement thereof since the claims introduce new matter not supported by the original disclosure. The original disclosure does not reasonably convey to a designer of ordinary skill in the art that applicant was in possession of the design now claimed at the time the application was filed. See *In re Daniels*, 144 F.3d 1452, 46 USPQ2d 1788 (Fed. Cir. 1998); *In re Rasmussen*, 650 F.2d 1212, 211 USPQ 323 (CCPA 1981).

Specifically, there is no support in the original disclosure for the following:

Independent claims 1,10 now includes the limitation "each of the plurality of mask fields being independently selectable as either write enabled mask field or write-disabled mask field". Claims 19, 26 now includes the limitation " each bit of a second operand is individually selectable as either having a first predetermined value or a second predetermined value. Claim 35 now includes the limitation " wherein each bit in the second operand is individually selectable as either having a first predetermined value or a second predetermined value". This rejection assumes the scope of the claims

provides for after the instruction is programmed the individually selecting each of the plural of fields or bits as write enabled or disabled.

These limitations are not described in the application as originally filed. The instructions of the application as originally filed do not provide for mask fields being individually selectable . Note, although claims 28 and 35 provide for operands fields the claims taken as a whole provide for these operands as mask fields. The instructions disclosed in the instant application as originally filed provide for some mask fields or bits which provided as either write enabled and others write disabled but the particular mask bits that are enabled or disabled are not selectable. For a particular instruction the same mask bits are always selected as respectively enabled or disabled the way the instruction was originally programmed. In the instant application as originally filed , there are no disclosed steps for, or mechanism for individually selecting the mask bits for a particular instruction. Therefore Claims 1, 10,19, 26 (and dependent claims 2-9,11-18,20-25,27-32) do not meet the description requirement of 112 first paragraph.

To overcome this rejection, applicant may attempt to demonstrate that the original disclosure establishes that he or she was in possession of the amended claims.

Claims 1-18 and 19-32 are rejected under 35 U.S.C. 112, first paragraph, because the specification, while being enabling for instructions that have mask bits , does not reasonably provide enablement for provide for mask fields of an instruction being individually selectable. Note that although the wording of claims 1,10, 19, and 26 provide for operand fields the claims as a whole provide for these operands as mask fields. The specification does not enable any person skilled in the art to which it

pertains, or with which it is most nearly connected, to provide and use instructions that include mask bits that are individually selectable the invention commensurate in scope with these claims. As disclosed the mask for each individual instruction cannot change. So the individual bits that are enabled or disabled do not change for an individual instruction. In order to make an use an instruction that additionally provides for individually selectable mask bits would have required alteration of the instructions as disclosed and a mechanism to alter the instruction and provide proper timing for altering and executing the instruction. This would have required undue experimentation to perform and therefore claims 1-32 do not meet the enablement requirement of 35 U.S.C. 112.

***Claim Rejections - 35 USC § 103***

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

**2.** Claims 1- 8,10-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hao (patent No. 4,569,016).

3. Hao taught the invention substantially as claimed including a data processing ("DP") system comprising (as to claim 1): A programmable processor comprising: data path an instruction path (path via instruction register to ALU and Mask and rotate logic in figured 2A and 2B); data path (path via general purpose registers and to mask and rotate logic and ALU in figures 2A and 2B);external interface (I-cache and D-

cache)operable to receive data from an external source (main memory which is external to the CPU 12) and communicate the received data over the data path (e.g. see figs. 1,2); register file (30) operable to receive and store data from the data path and communicate the data stored data to the (e.g., see figs. 2a,2b and col. 23, lines 49-59 and col. 24, lines 10-27),execution unit (42,56) coupled to the data path and operable to decode and executing instructions received from the instruction path(e.g., see col. 24, lines 10-51). Hao also taught in response to decoding a single processor instruction for writing data based on a mask and data contained in at least one register (e.g. see figs. 2a, 2b and col. 25,lines 23-66 and col. 26, lines 3-46), the mask comprising fields that each correspond to a field of the data contained in the at least one register(e.g., see col. 26,lines 36-46);the execution unit is operable to detect some of the fields of the mask as having a predetermined value to identify corresponding fields contained in the at least one register as write-enabled data fields(e.g., see col. 26, lines 26-46).

4. Hao did not expressly detail causing the write-enabled data fields to be written to a specified memory location. Hao however taught storing the write enabled data fields to a specified register (e.g., see col. 13, lines 7-12) and taught the processor performs rotate operations on data from a general purpose register and returns the result, or portion of the result to a general purpose register or to main storage (e.g., see col. 12, lines 42-60). Therefore one of ordinary skill would have been motivated to store the result of the mask and rotate to main storage. The Hao system stores results in main

storage and this would have reduced the number of registers needed to perform further processing.

5. Claim 1 includes the limitation of “each of the plurality of mask fields being independently selectable as either a write enabled mask field or a write disabled field”. The fields in the Hao teaching provide individual bits that individually provide for the enabling or disabling of bits of data for write in respective enable or disable status (e.g., see col. 13, lines 5-13). Hao taught a system where a substring within a string is selected using a leftmost index and rightmost index (e.g,. see col. 12, lines 8-29). The indexes comprise a series of bits that indicate the location and boundary within the string of the substring to be masked. The only limitation on the numbers within the indexes are that the rightmost does not exceed the leftmost in value which is an invalid situation. Therefore the situation where the leftmost and rightmost indexes are the same is within the teachings of Hao. This situation would provide a selection of any single bit with the mask string individually depending on what same number the leftmost and rightmost indexes comprised. Additionally Hao taught a field Bit 21 that indicated whether the selected subfield of the mask comprised a zero or one or correspondingly enabling or disabling write of the bit (or bits) of the operand (e.g., see col. 12, lines 8-19). Therefore The situation where the leftmost and rightmost indexes have the same value provide for a mask where Bit 21 indicates the value of an individually selected bit of the mask for selecting whether to enable or disable writing the corresponding single bit of the operand. Also Hao taught a system where any selected mask field (which can be one or more bits) can be enabled for write while the fields outside the mask can be

select as enabled or disabled; or where the selected mask field can be disabled while the fields outside the mask field can be enabled or disabled. Note that each of these combinations are provided for as Hao taught all zeros or all ones zeros surrounded by ones or ones surrounded by ones where the size of the selectively masked field is selected using the boundary location bits (the situation where the bits surrounding the masked bit are the same condition of enabled or disabled are provided for by enlarging the mask field).

6. As per claim 2, Hao taught each of the fields of the mask has a width of one bit (e.g., see col. 15, lines 39-44).

7. As per claim 3, Hao taught each of the fields of data contained in the at least one register has a width of one bit (e.g., see col. 15, lines 39-44).

8. As per claim 4, Hao writing comprises reading an unaltered field of data from the specified location and writing the unaltered field of data along with the write-enabled data fields of the specified location (e.g., see col. 15, lines 15-44). Therefore one when the operation was performed on data that originated from main memory and was later stored back to main memory the reading and writing of unaltered words to/from memory would have been performed in the Hao system (e.g., in the Hao system when the storage to storage instructions were used )(e.g., see col. 12, lines 21-54) .

9. As per claim 5, Hao taught the mask is contained in a specified register (e.g., see col. 28, lines 21-54).

10. As per claim 6, Hao taught storing the result a memory location (e.g., see col. 12, lines 42-60). Therefore one of ordinary skill would have been motivated to provide

the destination memory location in a register such as the instruction register when the destination address an immediate field or another register when the destination was specified using direct or indirect addressing.

11. As per claim 7, Hao taught the architecture comprising a 32-bit architecture (e.g., see col. 8, lines 64-68). Therefore one of ordinary skill would have been motivated to store the data or instructions to memory in specified memory locations comprises a section of memory having a specific width (e.g., 32-bits) and beginning at a specific memory address at least to allow later retrieval of stored data.

12. As per claim 8, Hao taught the predetermined logic value is 1(e.g. see col. 13, lines 6-15).

13. As per claim 10, Hao taught a data processing system comprising bus coupling components in the data processing system (e.g., see figs. 2a,2b) external memory (main memory that is external to the CPU) coupled to the bus (e.g., see figs, 1,2) a programmable microprocessor coupled to the bus and capable of operation independent of another host processor (e.g., see fig. 2) the microprocessor comprising an instruction path (path via instruction register to ALU and Mask and rotate logic in figured 2A and 2B); data path (path via general purpose registers and to mask and rotate logic and ALU in figures 2A and 2B);external interface (I-cache and D-cache)operable to receive data from an external source (main memory which is external to the CPU 12) and communicate the received data over the data path (e.g. see figs. 1,2); register file (30) operable to receive and store data from the data path and communicate the data stored data to the data path (e.g., see figs. 2a,2b and col.

23, lines 49-59 and col. 24, lines 10-27),execution unit (42,56) coupled to the data path and operable to decode and executing instructions received from the instruction path(e.g,. see col. 24, lines 10-51). Hao also taught in response to decoding a single processor instruction for writing data based on a mask and data contained in at least one register (e.g. see figs. 2a,2b and col. 25,lines 23-66 and col. 26, lines 3-46), the mask comprising fields that each correspond to a field of the data contained in the at least one register(e.g., see col. 26,lines 36-46);the execution unit is operable to detect some of the fields of the mask as having a predetermined value to identify corresponding fields contained in the at least one register as write-enabled data fields(e.g., see col. 26, lines 26-46).

14. Hao did not expressly detail causing the write-enabled data fields to be written to a specified memory location. Hao however taught storing the write enabled data fields to a specified register (e.g., see col. 13, lines 7-12) and taught the processor performs rotate operations on data from a general purpose register and returns the result, or portion of the result to a general purpose register or to main storage (e.g., see col. 12, lines 42-60). Therefore one of ordinary skill would have been motivated to store the result of the mask and rotate to main storage. The Hao system stores results in main storage and this would have reduced the number of registers needed to perform further processing.

15. Claim 10 includes the limitation of “each of the plurality of mask fields being independently selectable as either a write enabled mask field or a write disabled field”. The fields in the Hao teaching provide individual bits that individually provide for the

enabling or disabling of bits of data for write in respective enable or disable status (e.g., see col. 13, lines 5-13). Hao taught a system where a substring within a string is selected using a leftmost index and rightmost index (e.g,. see col. 12, lines 8-29). The indexes comprise a series of bits that indicate the location and boundary within the string of the substring to be masked. The only limitation on the numbers within the indexes are that the rightmost does not exceed the leftmost in value which is an invalid situation. Therefore the situation where the leftmost and rightmost indexes are the same is within the teachings of Hao. This situation would provide a selection of any single bit with the mask string individually depending on what same number the leftmost and rightmost indexes comprised. Additionally Hao taught a field Bit 21 that indicated whether the selected subfield of the mask comprised a zero or one or correspondingly enabling or disabling write of the bit (or bits) of the operand (e.g., see col. 12, lines 8-19). Therefore The situation where the leftmost and rightmost indexes have the same value provide for a mask where Bit 21 indicates the value of an individually selected bit of the mask for selecting whether to enable or disable writing the corresponding single bit of the operand. Also Hao taught a system where any selected mask field (which can be one or more bits) can be enabled for write while the fields outside the mask can be select as enabled or disabled; or where the selected mask field can be disabled while the fields outside the mask field can be enabled or disabled. Note that each of these combinations are provided for as Hao taught all zeros or all ones zeros surrounded by ones or ones surrounded by ones where the size of the selectively masked field is selected using the boundary location bits (the situation where the bits surrounding the

masked bit are the same condition of enabled or disabled are provided for by enlarging the mask field).

16. As per claim 11, Hao taught each of the fields of the mask has a width of one bit(e.g., see col. 15, lines 39-44).

17. As per claim 12, Hao taught each of the fields of the data contained in the at least one register has a width of one bit (e.g., see col. 15, lines 39-44).

18. As per claim 13, Hao writing comprises reading an unaltered field of data from the specified location and writing the unaltered field of data along with the write-enabled data fields of the specified location (e.g., see col. 15, lines 15-44). Therefore one when the operation was performed on data that originated from main memory and was later stored back to main memory the reading and writing of unaltered words to/from memory would have been performed in the Hao system. have been performed in the Hao system such as when the storage to storage instructions were used (e.g., see col. 12, lines 21-54) .

19. As per claim 14, Hao taught the mask is specified in a register (e.g., see col. 28, lines 21-54).

20. As per claim 15, Hao taught storing the result in a memory location (e.g., see col. 12, lines 21-60). Therefore one of ordinary skill would have been motivated to provide the destination memory location in a register such as the instruction register when the destination address an immediate field or another register when the destination was specified using direct or indirect addressing.

21. As per claim 16, Hao taught the architecture comprising a 32-bit architecture (e.g., see col. 8, lines 64-68). Therefore one of ordinary skill would have been motivated to store the data or instructions to memory in specified memory locations comprises a section of memory having a specific width (e.g., 32-bits) and beginning at a specific memory address at least to allow later retrieval of stored data.

.

22. As per claim 17, Hao taught the predetermined value is 1(e.g. see col. 13, lines 6-15).

23. Claims 9,18, are rejected under 35 U.S.C. 103(a) as being unpatentable over Hao applied to claims 1-8,10 above, and further in view of Kabir (patent No. 5,933,160).

24. As per claim 9, 18 Kabir taught in response to decoding a second single instruction specifying a register containing a plurality of floating point operands and another register containing a second plurality of floating point operands ; multiplying the plurality of floating point operands in the register by the plurality of operands to produce a plurality of products; and providing the partitioned field of a result as a concatenated result (e.g., see fig. 4, 5a, 5b and col. 8,lines 21-45).

25. It would have been obvious to one of ordinary skill to combine the teachings of Hao and Kabir. Both references were directed toward performing operations of partial widths of data stored in registers. Kabir taught further operations to be performed on the partial width data such as multiplication on floating point data (e.g., see fig. 4) for performing image processing in a digital system (e.g., see col. 1, lines 6-11) consequently one of ordinary skill would have been motivated to incorporate the floating

point operations to the Hao system at least to provide the capability use in addition applications such as graphics applications.

26. Claims 19-23,26-30, are rejected under 35 U.S.C. 103(a) as being unpatentable over Hao.

27. As per claims 19,26 Hao taught a programmable processor comprising: : data path an instruction path (path via instruction register to ALU and Mask and rotate logic in figured 2A and 2B); data path (path via general purpose registers and to mask and rotate logic and ALU in figures 2A and 2B);cache (I-cache and D-cache)operable to retain data from an external interface and data path (e.g. see figs. 1,2); register file (30) operable to receive and store data from the data path and communicate the data stored data to the (e.g., see figs. 2a,2b and col. 23, lines 49-59 and col. 24, lines 10-27). Hao taught execution unit coupled to the instruction and data paths that is operable to execute instructions received from the instruction path,(e.g. see figs. 2a,2b and col. 25,lines 23-66 and col. 26, lines 3-46), and performing a bitwise insert operation operating on a first operand and a second operand stored in the at least one register (e.g., see col. 26, lines 36-46); and for each bit in the first operand, the bitwise insert operation inserting the bit into a corresponding bit position in a destination value if a corresponding bit in the second operand has a first predetermined value (e.g., see col. 15, lines 39-44). As to the external interface Hao taught a bus units that interface system bus and other buses that transfer data to /from the CPU via the cache and which are external to the CPU (12)(e.g., see fig. 1). Also one of ordinary skill would have been motivated to store data in the interfaces at least to provide flexlible timing of

the transfer of data between the system bus or other external buses and the instruction bus , internal bus or memory bus. Also, Hao did not expressly detail a virtual memory addressing unit. However as to the virtual addressing unit the use of virtual addressing to a memory was well known in the art at the time of the claimed invention allowing the system to more efficiently address memory when plural tasks were being processed. Therefore one of ordinary skill would have been motivated to incorporate a virtual memory addressing means at least to allow for the simultaneous processing of multiple tasks while simplifying the addressing of storage locations.

28. Claims 19, and 26 include the limitation: "each bit in the second operand is individually selectable as either having a first predetermined value or a second predetermined value and for each bit in the first operand, the bitwise insert operation inserting the bit into a corresponding bit position in a destination value if a corresponding bit in the second operand has the first predetermined value" The fields in the Hao teaching provide individual bits that individually provide for the enabling or disabling of bits of data for write in respective enable or disable status (e.g., see col. 13, lines 5-13). Hao taught a system where a substring within a string is selected using a leftmost index and rightmost index (e.g,. see col. 12, lines 8-29). The indexes comprise a series of bits that indicate the location and boundary within the string of the substring to be masked. The only limitation on the numbers within the indexes are that the rightmost does not exceed the leftmost in value which is an invalid situation. Therefore the situation where the leftmost and rightmost indexes are the same is within the teachings of Hao. This situation would provide a selection of any single bit with the

mask string individually depending on what same number the leftmost and rightmost indexes comprised. Additionally Hao taught a field Bit 21 that indicated whether the selected subfield of the mask comprised a zero or one for correspondingly enabling or disabling write of the bit ( or bits) of the operand (e.g., see col. 12, lines 8-19). Therefore The situation where the leftmost and rightmost indexes have the same value provide for a mask where Bit 21 indicates the value of an individually selected bit of the mask for selecting whether to enable or disable writing the corresponding single bit of the operand (the situation where the bits surrounding the masked bit are the same condition of enabled or disabled are provided for by enlarging the mask field).

29. As per claim 20,27, Hao taught the first predetermined value is a logic 1 (e.g. see col. 13, lines 6-15).

30. As per claim 21,28 Hao taught for each bit in the first operand, a corresponding bit position in the destination value is maintained as unchanged if a corresponding bit in the second operand has a second predetermined value (e.g., see col. 15, lines 39-44).

31. As per claim 22,29, Hao taught the second predetermined value is 0 (e.g. see col. 13, lines 6-15).

32. As per claim 23,30, Hao taught the destination value is stored into memory (e.g., see col. 12, lines 42-60).

33. Claims 24,25,31,32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hao as applied to claims 19,26 above, and further in view of Kabir.

34. As per claims 24,31 Kabir taught arithmetic operation where operands were stored in registers of 64-bit width (e.g., see col. 8, lines 5-37 and col. 9, line 27-col. 10,

line 28 and fig.5A,5B). As to the operand being 64 bit width one of ordinary skill would have been motivated to use data with more bits such as 64, 128 etc to take advantage of the increasing capacity of industry standard memories, data paths and processors at the time of the claimed invention.

35. It would have been obvious to one of ordinary skill to combine the teachings of Hao and Kabir. Both references were directed toward performing operations of partial widths of data stored in registers. Kabir taught further operations to be performed on the partial width data such as multiplication on floating point data (e.g., see fig. 4) for performing image processing in a digital system (e.g., see col. 1, lines 6-11) consequently one of ordinary skill would have been motivated to incorporate the floating point operations to the Hao system at least to provide the capability use in addition applications such as graphics applications.

36. As per claim 25, Kabir taught instructions further comprises a plurality of different group floating-point arithmetic operations that arithmetically operate one multiple floating-point operands stored in partitioned fields of an operand register in the plurality of registers to produce a concatenated result that is returned to a register in the plurality of registers, wherein the concatenated result comprises a plurality of individual floating-point results (e.g., see col. 4, lines 23-61 and col. 7, line 7-col. 8, line 41).

37. As per claim 32, Kabir taught executing a plurality of different group floating point operation that arithmetically operate on multiple floating point operands partitioned in fields of an operand register in the plurality of registers to produce a concatenated result that is returned to a register in the plurality of registers, wherein the concatenated result

comprises a plurality of individual floating point results (e.g., see fig. 4, 5a, 5b and col. 8,lines 21-45 and col. 5, lines 6-47 and col. 4, lines 23-64).

## *Double Patenting*

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

Claims 1-32 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 1-9, 28-34 of copending Application No. 10/757866. Although the conflicting claims are not identical, they are not patentably distinct from each other because as shown by the side by side listing of the claims the claims of the instant application and SN 10/757866 are substantially similar. The portion of the claims independent claims 1,10,19,26 of the instant application not specifically detailed in the method claims of SN 10/757866 include specifics of a programmable processor which are well known in the art with

respect to processors. One of ordinary skill would have been motivated in incorporate these well known features (such as external interface, data and instruction paths and implementation as microprocessor independent of a host and memory hierarchy including a cache memory, and virtual memory addressing unit ) of programmable processors at the time of the claimed invention at least to provide the functionality of the programmable processor performing the method in SN 10/757866 and steps of the method in SN 10/757866.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

| Instant application | SN 10757866 |
|---|---|
| 1. A programmable processor comprising: an instruction path; a data path; and external interface operable to receive data from an external data path and communicate the stored data to the data path; and an execution coupled to the instruction and data paths and operable to decode and execute instructions received from the instruction path, wherein in response to decoding a single instruction for writing data to memory based on a mask and data contained in at least one | 1. A method for processing data using programmable processor comprising: decoding a single instruction for writing to memory based on a mask and data contained in at least one register, the mask comprising a plurality of mask fields that each corresponds to a data field of the data contained in the at least one register, each of the plurality of mask fields being independently selectable as either a write-enabled mask field or a write-disabled mask field; detecting some of the mask |

| | |
|---|---|
| register, the mask comprising a plurality of mask fields that each correspond to a data | fields as being selected as write enable mask fields to identify corresponding data |
| Instant application | SN 10/757866 |
| field of the data contained in the at least one register, each of the plurality of mask fields being independently selectable as either a write-enabled mask field or a write-disabled mask field, the execution unit is operable to: (i) detect some of the mask fields of the mask as being selected as write enabled mask operable to identify corresponding data fields of the data contained in the at least one register as write-enable data fields; and (ii) cause the write enabled data fields to be written to a specified memory location. | fields of the data contained in the at least one register as write-enabled data fields; and writing the write-enabled data fields. |
| 2. The processor of claim 1 wherein each of the mask fields of the mask has a width of one bit. | 2. The method of claim 1 wherein each of the mask fields of the mask has a width of one bit. |
| 3.The processor of claim 1 wherein each of the data fields of the data contained in the at least one register has a width of one | 3.The method of claim 1 wherein each of the data fields of the data contained in the at least one register has a width of one bit. |

| | |
|---|---|
| bit. | |
| Instant application | SN 10/757866 |
| 4. The processor of claim 1 wherein the execution unit is operable to cause the write-enabled data fields to be written to the specified memory location by reading an unaltered field of data from the specified memory location and writing the unaltered field of data along with write-enable data fields to the specified location. | 4. the method of claim 1 wherein the writng step further comprisies reading unaltered field of data form the specified memory location and writing the unaltered field of data along with the write-enabled data fields to the specified memory location. |
| 5. The processor of claim 1 wherein the mask is contained in a specified register. | 5. The method of claim 1 wherein the mask is contained in a specified register |
| 6. The processor of claim 1 wherein the memory location is specified by a register. | 6. The method of claim 1 wherein the memory location is specified by a register |
| 7. The processor of claim 1 wherein the specified memory location comprises a section of memory having a specified width and beginning at a specified memory address. | 7. The method of claim 1 wherein the specified memory location comprises a section of memory having a specified width and beginning at a specified memory address. |
| 8. The processor of claim 1 wherein each write -enabled mask field is indicated as a | 8. The method of claim 1 wherein each write -enabled mask field is indicated as a |

| | |
|---|---|
| logic 1. | logic 1. |
| Instant application | SN 10/757866 |
| 9. The processor of claim 1 wherein the execution unit is further operable to, in response to decoding a second single instruction specifying a register containing a first plurality of floating-point operands and another register containing a second plurality of floating point operands, multiply the first plurality of floating point operands by the second plurality of floating point operands to produce a plurality of products and provide the plurality of products to partitioned fields of a result register as a catenated result. | 9. The method of claim 1 further comprising decoding a second single instruction specifying a register containing a first plurality of floating point operands and another register containing a second plurality of floating point point operands; multiplying the first plurality of floating point operands by the second plurality of floating point operands to produce a plurality of products; and prodiving of products to parititioned fields of a result register as a catenated result. |
| 10. A data processing system comprising:(a) a bus coupling components in the data processing system; (b) and external memory coupled to the bus; (c) a programmable microprocessor to the bus | 1. A method for processing data using programmable processor comprising: |

| | |
|---|---|
| and capable of operation independent of | |
| Instant application | SN 10/757866 |
| another host processor, the microprocessor comprising: an instruction path; a data path; an external interface operable to receive data from an external source and communicate data over the data path; a register file operable to receive and store data from the data path and communicate the stored data to the data path and an execution unit coupled to the instruction and data paths and operable to decode and execute instructions received from the instruction path, wherein in response to decoding a single instruction for writing data to memory based on a mask an data contained in at least one register, the mask comprising a plurality of mask fields that each correspond to a data field of the data contained in the at least one register, | decoding a single instruction for writing to memory based on a mask and data contained in at least one register, the mask comprising a plurality of mask fields that each corresponds to a data field of the |

| | |
|---|---|
| each of the plurality of mask fields being independently selectable as either a write- | data contained in the at least one register, each of the plurality of mask fields being |
| Instant application | SN 10/757866 |
| enabled mask field or a write-disabled mask field the execution unit is operable to: (i) detect some of the fields of the mask selected as write-enabled mask fields to identify corresponding data fields of the data contained in the at least one register as write-enabled data fields; and (ii) cause the write-enabled data fields to be written to a specified memory location. | independently selectable as either a write-enabled mask field or a write-disabled mask field; detecting some of the mask fields as being selected as write enable mask fields to identify corresponding data fields of the data contained in the at least one register as write-enabled data fields; and writing the write-enabled data fields. |
| 11. The system of claim 10 wherein each of the mask fields of the mask has a width of one bit. | 2. The method of claim 1 wherein each of the mask fields of the mask has a width of one bit. |
| 12.The system of claim 10 wherein each of the data fields of the data contained in the at least one register has a width of one bit. | 3.The system of claim 1 wherein each of the data fields of the data contained in the at least one register has a width of one bit. |
| 13. The system of claim 10 wherein the execution unit is operable to cause the write-enabled data fields to be written to the specified memory location by reading | 4.. The system of claim 1 wherein the execution unit is operable to cause the write-enabled data fields to be written to the specified memory location by reading |

| | |
|---|---|
| an unaltered field of data from the specified memory location and writing the unaltered field of data along with write-enable data fields to the specified location. | an unaltered field of data from the specified memory location and writing the unaltered field of data along with write-enable data fields to the specified location. |
| 14. The system of claim 10 wherein the mask is contained in a specified register. | 5.The method of claim 1 wherein the mask is contained in a specified register. |
| 15. The system of claim 10 wherein the memory location is specified by a register. | 6. the method of claim 1 wherein the memory location is specified by a register. |
| 16. The system of claim 10 wherein the specified memory location comprises a section of memory having a specified width and beginning at a specified memory address. | 7. The method of claim 1 wherein the specified memory location comprises a section of memory having a specified width and beginning at a specified memory address. |
| 17.. The system of claim 10 wherein each write -enabled mask field is indicated as a logic 1. | 8.The method of claim 1 wherein each write-enabled mask field is indicated as a logic 1. |
| 18. The system of claim 10 wherein the execution unit is further operable to, in response to decoding a second single instruction specifying a register containing a plurality of floating point operands and another register containing a second | 9, the method of clam 1 further comprising decoding a second single instruction specifying a register containing a first plurality of floating point operands and another register containing a second |

| | |
|---|---|
| plurality of floating point operands, multiply the first plurality of floating point operands by the second plurality of floating point operands to produce a plurality of products to provide the plurality of products to partitioned fields of a result register as a concatenated result. | plurlaity of floating point operands; multiplying the first plurality of floatiing-point operands by a second plurality of floating-point operands to produce a plurality o product to partitioned fields of a result register as a catenated result. |
| 19. A programmable processor comprising: a virtual memory addressing unit; an instruction path and a data path; an external interface operable to receive data from an external source and communicate the received data over the data path; a cache operable to retain communicated between the external interface and the data path; a register path comprising a plurality of registers coupled to the data path; and an execution unit, coupled to the instruction and data paths that is operable to decode and execute instructions received from the instruction path the execution unit capable | 28 a method for processing data in a programmable processor the method comprising: Decoding a single instruction for performing a bitwise insert operation on data in at least one register in a register file within a register file within a programmable processor, the bitwise |

| | |
|---|---|
| of performing a bitwise insert operation that operates on a first and a second operand stored in at least one register file, wherein each bit in the second operand is individually selectable as either having a first predetermined value or a second predetermined value wherein each bit in the first operand, the bitwise insert operation inserts the bit into a corresponding bit position in a destination value if a corresponding bit in the second operand has a first predetermined value. | insert operation operating on a first operand and a second operand stord in at least one register in the register file, wherein each bit in the second operand is individually selectable as either having a first predetermined value or a second predetermined value; and for each bit in the first operand, the bitwise insert operation inserting the bit into a corresponding bit position in a destination value if a corresponding bit in the second operand has the first predetermined value. |
| 20. The programmable processor of claim 19 wherein the first predetermined value is a logic 1. | 29.The method of claim 28 wherein the first predetermined value is a logic 1. |
| 21. The programmable processor of claim 19 wherein for each bit in the first operand, the bitwise insert operation maintains a corresponding bit position in the destination value as unchanged if a corresponding bit in the second operand has the second predetermined value. | 30. The method of claim 28 wherein for each bit in the first operand, a corresponding bit position in the destination value is maintained as unchanged if a corresponding bit in the second operand has the second predetermined value. |

| | |
|---|---|
| 22. . The programmable processor of claim 21 wherein the second predetermined value is 0. | 31. the method of claim 30 wherein the second predetermined value is a logic 0. |
| Instant application | SN 10/757866 |
| 23. . The programmable processor of claim 19 wherein the bitwise insert operation stores operands the destination value into memory.. | 32. The method of claim 28 further comprising a step of storing the destination value into memory, |
| 24. . The programmable processor of claim 19 wherein each of the first and second operands has a width of 64 bits. | 33. The method of claim 28 wherein each of the first and second operands has a width of 64 bits. |
| 25. . The programmable processor of claim 19 wherein the execution unit is further capable of executing a plurality of different group floating-point arithmetic operations that arithmetically operate on multiple floating-point operands stored in partitioned fields of registers in the register file to produce a catenated result that is returned to a register in the register file, wherein the catenated result comprises a | 34. The method of claim 28, further comprising a step of executing a plurality of different group floating point arithmetic operations arithmetically operate on multiple floating-point operands stored in partitioned fields of registers in the register file to produce a catenated result that is returned to a register in the reigster file, wherein the catenated result comprises a plurality of |

| | |
|---|---|
| plurality of individual floating point results. | individual floating-point results. |
| Instant application | SN 10/757866 |
| 26. A device having installed therein a programmable processor comprising: a virtual memory addressing unit; an instruction path and a data path; an external interface operable to receive data from an external source and communicate the received data over the data path; a cache operable to retain communicated between the external interface and the data path; a register path comprising a plurality of registers coupled to the data path; and an execution unit, coupled to the instruction and data paths that is operable to decode and execute instructions received from the instruction path the execution unit capable of performing a bitwise insert operation that operates on a first and a second operand stored in at least one register file, wherein each bit in | 28 a method for processing data in a programmable processor the method comprising: Decoding a single instruction for performing a bitwise insert operation on data in at least one register in a register file within a register file within a programmable processor, the bitwise insert operation operating on a first operand and a second operand stord in at least one register in the register file, wherein each bit in the second operand is |

| | |
|---|---|
| the second operand is individually selectable as either having a first predetermined value or a second predetermined value wherein each bit in the first operand, the bitwise insert operation inserts the bit into a corresponding bit position in a destination value if a corresponding bit in the second operand has a first predetermined value. | individually selectable as either having a first predetermined value or a second predetermined value; and for each bit in the first operand, the bitwise insert operation inserting the bit into a corresponding bit position in a destination value if a corresponding bit in the second operand has the first predetermined value. |
| 27. . The device of claim 26 wherein the first predetermined value is a logic 1. | 29.The method of claim 28 wherein the first predetermined value is a logic 1. |
| 28. The device of claim 26 wherein for each bit in the first operand, the bitwise insert operation maintains a corresponding bit position in the destination value as unchanged if a corresponding bit in the second operand has the second predetermined value. | 30. The method of claim 28 wherein for each bit in the first operand, a corresponding bit position in the destination value is maintained as unchanged if a corresponding bit in the second operand has the second predetermined value. |
| 29. . The device of claim 28 wherein the second predetermined value is 0. | 31. the method of claim 30 wherein the second predetermined value is a logic 0. |
| 30. The device of claim 26 wherein the | 32. The method of claim 28 further |

| | |
|---|---|
| bitwise insert operation stores the destination value into memory. | comprising a step of storing the destination value into memory, |
| Instant application | SN 10/757866 |
| 31. The device of claim 26 wherein each of the first and second operands has a width of 64 bits. | 33. The method of claim 28 wherein each of the first and second operands has a width of 64 bits. |
| 32 The device of claim 26 wherein the execution unit is further capable of executing a plurality of different group floating-point arithmetic operations that arithmetically operate on multiple floating-point operands stored in partitioned fields of registers in the register file to produce a catenated result that is returned to a register in the register file, wherein the catenated result comprises a plurality of individual floating point results. | 34. The method of claim 28, further comprising a step of executing a plurality of different group floating point arithmetic operations arithmetically operate on multiple floating-point operands stored in partitioned fields of registers in the register file to produce a catenated result that is returned to a register in the register file, wherein the catenated result comprises a plurality of individual floating-point results. |

## ***Response to Arguments***

Applicant's arguments filed 4/30/08 have been fully considered but they are not persuasive.

The applicant argues in substance that the Hao reference did not teach the claimed features of "each of the plurality of mask fields being independently selectable as either a write-enabled mask field or a write disabled mask field; and wherein each bit in the second operand is individually selectable as either having a first predetermined value or a second predetermined value. As to this argument these features are taught by Hao as discussed in the outstanding rejection above.

The applicant also argues that the Kabir reference does not qualify as prior art. The Examiner contends that the Kabir indeed does qualify as prior art as the parent case does not provide a teaching or enabling disclosure of the claimed features. The declaration that is referred to by the applicant does not specifically address the claimed features and additionally the claimed features were not part of the claims in the instant application when the declaration was filed. The instructions at the time the declaration was filed referred the claimed instruction that did not require independently selectable mask fields and instead for each instruction the particular disclosed mask was predetermined. Consequently the declaration does not provide support for the feature satisfying the 112 first paragraph requirements as detailed in the outstanding 112 first paragraph rejection above.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Eric Coleman whose telephone number is (571) 272-4163. The examiner can normally be reached on Monday-Thursday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

EC

/Eric Coleman/
Primary Examiner, Art Unit 2183